Abschlussbericht

Für das ZIM-Kooperationsprojekt mit dem Förderkennzeichen KF2421105BZ3

INSA

Integrierte softwaregestützte
Sicherheitsanalyse von
Automatisierungsanlagen

Zuwendungsgeber	Bundesministerium für Wirtschaft und Energie Scharnhorststr. 34-37, 10115 Berlin
Projektträger	AiF Arbeitsgemeinschaft industrieller Forschungsvereinigungen „Otto von Guericke“ e.V. Bayenthalgürtel 23, 50968 Köln
Laufzeit des Vorhabens	Juni 2013 bis November 2015

Christopher Tebbe, Matthias Glawe, Karl-Heinz Niemann, Alexander Fay, Josha Dittgen, Jörg Eckardt,

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

22.02.2016

- Überarbeitete und gekürzte Fassung zur Veröffentlichung -

Projektpartner

Hochschule Hannover	HOCHSCHULE HANNOVER UNIVERSITY OF APPLIED SCIENCES AND ARTS – *Fakultät I Elektro- und Informationstechnik*
Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg	HELMUT SCHMIDT UNIVERS Universität der Bundes AT
M&M Software GmbH	M&M
ConSecur GmbH	ConSecur security & consulting
PHOENIX CONTACT GmbH & Co. KG	PHOENIX CONTACT
Niedersächsisches Ministerium für Inneres und Sport - Wirtschaftsschutz	Niedersächsisches Ministerium für Inneres und Sport

Ansprechpartner

Hochschule Hannover
Fakultät I – Elektro- und Informationstechnik
Fachgebiet Prozessinformatik/Automatisierungstechnik
Ricklinger Stadtweg 120
30459 Hannover

E-Mail: Karl-Heinz.Niemann@HS-Hannover.de

Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
Professur für Automatisierungstechnik
Holstenhofweg 85
22043 Hamburg

E-Mail: Alexander.Fay@hsu-hh.de

Inhaltsverzeichnis

Inhaltsverzeichnis 3
Abbildungsverzeichnis 4
Tabellenverzeichnis 4
1 Voruntersuchungen (AP 2) 5
2 Anforderungsanalyse (AP 3) 6
3 Grobentwurf der Systemarchitektur (AP 4) 7
3.1 Vorstellung Web-Service 8
3.2 Vorstellung Client / Benutzeroberfläche 9
3.3 Vorstellung logischer Ablauf der Hauptfunktionen 10
4 Erfassen der Schwachstellen und Bedrohungen (AP 5) 13
5 Erstellung Funktionsmuster entsprechend Grobentwurf der Systemarchitektur (AP 7) 19
6 Import der Projektierungsdaten aus Engineering-Werkzeug (AP 8) 19
7 Problemlösungskomponente (AP 10) 21
8 Erstellung der Wissensbasis (Zusammenfassung des gesamten Strukturwissens) (AP 11) 23
8.1 Aufbau der Wissensbasis 23
8.2 Verwaltung der Wissensbasis 25
9 Benutzeroberfläche (AP 12) 27
10 Gesamtintegration des Funktionsmusters (AP 14) 28
11 Qualitätssicherung (AP 15) 28
12 Test und Verifikation des Prototyps (AP 16) 28
13 Vergleich der angestrebten und erreichten technischen Parameter 30
13.1 Funktionale Anforderungen 30
13.2 Erläuterungen zu den nicht (vollständig) erfüllten funktionalen Anforderungen 30
13.3 Nicht funktionale Anforderungen 32
13.4 Erläuterungen zu den nicht (vollständig) erfüllten nicht-funktionalen Anforderungen 32
14 Schlusswort 33
15 Veröffentlichungen 34
16 Literatur 35

Abbildungsverzeichnis

Abbildung 1: Darstellung der Schnittstellen zwischen Web-Service und Benutzeroberfläche (GUI) 9
Abbildung 2: Grober Aufbau der Client-Seite..... 9
Abbildung 3: Verwendete Technologien der Client-Seite 10
Abbildung 4: Beispiel für das Anlegen eines Projekts und den Import von Anlageninformationen..... 11
Abbildung 5: Beispiel für die Vorgehensweise bei der Durchführung einer IT-Sicherheitsanalyse..... 11
Abbildung 6: Beispiel für die Vorgehensweise für die Bearbeitung des IT-Sicherheitswissens..... 12
Abbildung 7: Strukturierung der wesentlichen Eigenschaften eines Assets 13
Abbildung 8: Strukturierung der Bedrohungen..... 14
Abbildung 9: Übersicht der ermittelten Maßnahmen. 15
Abbildung 10: Verteilung des Wissens im Funktionsmuster..... 23
Abbildung 11: Übersicht über den gesamten Aufbau der Wissensbasis 24
Abbildung 12: Abstraktion der OWL-Ontologie zur Laufzeit des Funktionsmusters 26

Tabellenverzeichnis

Tabelle 1: Einstufung des Implementierungsaufwands für Schutzmaßnahmen..... 16
Tabelle 2: Wirksamkeitsstufen von Schutzmaßnahmen 16
Tabelle 3: Einstufung der im Rahmen der Bedrohungsanalyse ermittelten Schutzmaßnahmen 17
Tabelle 4: Einstufung von Sicherheitsfunktionalitäten 18
Tabelle 5: Einstufung von Sicherheitsprozessen 18
Tabelle 6: Ermittlung des Protection-Levels anhand der SL und ML..... 18
Tabelle 7: Auflistung aller Regelklassen und ihrer Bedeutung..... 25
Tabelle 8: Übersicht über alle (nicht) umgesetzten funktionalen Anforderungen 30
Tabelle 9: Erläuterungen zu nicht umgesetzten Anforderungen bei Funktionsmusterfunktionen 30
Tabelle 10: Erläuterungen nicht umgesetzte Anforderungen an Informationsgewinnschnittstellen .. 31
Tabelle 11: Erläuterungen zu den nicht umgesetzten Anforderungen an die Speicherung von Daten 31
Tabelle 12: Übersicht über alle (nicht) umgesetzten nicht-funktionalen Anforderungen..... 32
Tabelle 13: Erläuterungen zu den nicht umgesetzten Entwicklungsrestriktionen..... 32

1 Voruntersuchungen (AP 2)

Im Vorlauf des Projekts sowie in regelmäßigen Abständen während der kompletten Projektphase wurden Recherchen zur Ermittlung des aktuellen Stands der Technik im Bereich der Software-Unterstützung von IT-Sicherheitsanalysen automatisierungstechnischer Anlagen durchgeführt.

Bereits zum Start des Projekts bestanden einige Lösungen für softwaregestützte IT-Sicherheitsanalysen. Diese sind jedoch stark auf das Einsatzfeld der Standard-IT begrenzt und bedürfen der manuellen Eingabe der Systemstruktur durch den Anwender. Die Implementierung von Automatisierungskomponenten ist möglich, je nach Größe der Automatisierungsanlage jedoch mit großem Aufwand verbunden. Werkzeuge zur Schwachstellen- bzw. Bedrohungsanalyse geben unter anderem Aufschluss über mögliche IT-Sicherheitsmaßnahmen, greifen jedoch oftmals aktiv in das Netzwerk ein, was in der Automatisierungstechnik nicht akzeptiert wird. Auch existieren Lösungen auf Basis wissensbasierter Methoden, die jedoch sehr unflexibel sind.

Während des Projekts zeigte sich, dass parallel zu INSA weitere Werkzeuge für IT-Sicherheitsanalysen automatisierungstechnischer Anlagen entwickelt wurden. Diese bieten im Hinblick auf die Aufwandsreduzierung von Bedrohungsanalysen nur geringe Vorteile bzw. sollen nur als Einstieg dienen. Ein Beispiel ist Light and Right Security (LARS) [1], das im November 2014 vom Bundesamt für Sicherheit in der Informationstechnik (BSI) veröffentlicht wurde und vor allem im deutschen Raum von Interesse ist. Das Ziel von LARS ist es, besonders KMU an die Problematik der IT-Sicherheit von Anlagen heranzuführen und ein Schutzkonzept zu entwickeln. Ein anderes Werkzeug ist das Cybersecurity Evaluation Tool (CSET) [2] vom amerikanischen ICS-Cert [3]. Im Gegensatz zu LARS kann bei CSET auch die zu betrachtende Anlage modelliert werden. Bei beiden Werkzeugen muss ein umfangreicher Fragekatalog abgearbeitet werden. Daraus wird ein Ergebnis für die analysierte Anlage abgeleitet und Schutzmaßnahmen vorgeschlagen. Weitere Informationen gehen nicht in die Analysen ein. Die Modellierung sowie Beantwortung der Fragen muss komplett durch die bedienende Person geschehen. Mit diesen Werkzeugen ist ein erster Schritt für eine Softwareunterstützung von IT-Sicherheitsanalysen getan. Es bestehen jedoch noch viele Möglichkeiten, den Aufwand und das benötigte Wissen für IT-Sicherheitsanalysen zu reduzieren.

2 Anforderungsanalyse (AP 3)

Kurz nach Beginn des Projekts wurde die Erarbeitung des Lastenhefts durch alle Projektpartner begonnen. Das Lastenheft stellt eine Verfeinerung der technischen Zielstellung aus dem Projektantrag dar und enthält alle Anforderungen (funktional und nicht-funktional), die an das Funktionsmuster gestellt werden. Außerdem wurde die Umgebung des Funktionsmusters festgelegt, alle zu betrachtenden Use-Cases definiert und die Ziele des Projekts festgelegt. Das Lastenheft wurde durch mehrfache Reviews durch die Projektpartner überprüft und von allen Projektpartnern gemeinsam in einer finalen Version verabschiedet. Die einzelnen Bestandteile des Lastenhefts sind:

- Beschreibung der Ausgangsituation inklusive der Ziele und Risiken
- Beispielanlagenstruktur an der die Bedrohungsanalyse aus AP 5 durchgeführt wird
- Festlegung des grundsätzlichen Aufbaus des Funktionsmusters
- Definition des Funktionsumfangs durch Use-Cases und (nicht) funktionale Anforderungen

Zusätzlich zum Lastenheft wurde in diesem Arbeitspaket ein Fragebogen erstellt, der an kleine und mittelständische Unternehmen gerichtet ist, die automatisierungstechnische Anlagen betreiben. Diese Unternehmen entsprechen der Zielgruppe des Projekts. Die Fragen sind dabei in verschiedene Kategorien eingeteilt. Neben allgemeinen Fragen zu der befragten Person wurden deren Einschätzungen und Erfahrungen bezüglich Bedrohungen, Schwachstellen und Schutzmaßnahmen für automatisierungstechnische Anlagen abgefragt. Auch der bisherige Umgang mit und das Verständnis von IT-Sicherheitsvorgehensmodellen wurden ermittelt. Eine Auswertung ergab einige Hinweise, welche Funktionalitäten aus Sicht der Betreiber einer automatisierungstechnischen Anlage wünschenswert sind. Dies sind zum Beispiel der Umfang möglicher Assets und welche Anlageninformationen bei einer IT-Sicherheitsanalyse zu berücksichtigen sind sowie der Bedarf, ständig über aktuelles IT-Sicherheitswissen zu verfügen. Zudem lies sich feststellen, dass ein Bedarf bei der Unterstützung im Bereich der IT-Sicherheit automatisierungstechnischer Anlagen besteht. Die Ergebnisse des Fragebogens flossen in die Entwicklung des Funktionsmusters und dessen Wissensbasis ein.

3 Grobentwurf der Systemarchitektur (AP 4)

Aufbauend auf den Ergebnissen aus AP3 wurde durch die Projektpartner ein Grobentwurf der Systemarchitektur durchgeführt, der die Entwicklung und Umsetzung der Systemarchitektur inkl. der Spezifikation der Funktionsmusterkomponenten und der Schnittstellen beschreibt.

Der Grobentwurf wurde in zwei Dokumente aufgeteilt, das Pflichtenheft und ein Architektur-Dokument. Das Pflichtenheft referenziert auf Teile des Lastenhefts und stellt den logischen Aufbau des Funktionsmusters und die Realisierungsfestlegungen für die Funktionen des Funktionsmusters dar. Zudem wurden basierend auf den Use-Cases und funktionalen Anforderungen des Lastenhefts Nutzer-Kontext-Szenarien entwickelt, welche diese zusammenfassen und in Beziehung zueinander bringen. Die Entwicklung der Nutzer-Kontext-Szenarien erfolgte teilweise im Rahmen von AP 12 und AP 4. Einflüsse durch nicht funktionale Anforderungen wurden ebenfalls erfasst. Im Pflichtenheft wurden die Nutzer-Kontext-Szenarien um die Festlegungen erweitert, wie die Anforderungen und Funktionen während der Implementierung realisiert werden sollen.

Das Pflichtenheft umfasst die folgenden Bestandteile:

- Beschreibung der Ausgangsituation inklusive aktualisierter Ziele und Risiken
- Zielbestimmungen (Auflistung der Muss-, Wunsch- und Abgrenzungskriterien)
- Systemkontext inkl. Anwendergruppen und Anwendungsumgebung
- Festlegungen von Rahmenbedingungen
- Domänenmodell des Funktionsmusters
- Aktualisierung der (nicht) funktionalen Anforderungen an den aktuellen Stand
- Zusammenführung der Use-Cases und funktionalen Anforderungen aus dem Lastenheft zu Nutzer-Kontext-Szenarien (siehe oben).
- Datenmodell des Funktionsmusters
- Auswahl eines regelbasierten Systems als wissensbasiertes System für das Funktionsmuster
- Beschreibung der logischen Komponenten und Schnittstellen
- Designfestlegungen für die Benutzeroberfläche

Detaillierte Architekturentscheidungen wurden in ein eigenes Architektur-Dokument ausgegliedert. Eine Grundversion des Architektur-Dokuments wurde zum Ende des Arbeitspakets AP 4 festgelegt. Im Gegensatz zum Pflichtenheft wurde das Architektur-Dokument nicht finalisiert, sondern im Laufe des Projektes fortgeschrieben und zum Projektende finalisiert. Hiermit wurde den zu erwartenden Forschungsergebnissen Rechnung getragen, welche Einfluss auf die Systemarchitektur haben und erst im Rahmen der weiteren Arbeitspakete erarbeitet werden konnten. Zusätzlich diente das Architekturdokument zur Dokumentation der entwickelten Software für die weitere Verwendung nach Projektabschluss.

Das Architektur-Dokument hat folgende Bestandteile:

- Definition der Schnittstellen und Komponenten
- Beschreibungen aller relevanten Klassen
- Darstellung des dynamischen Verhaltens mittels UML-Diagrammen
- Festschreibung der Entwurfsentscheidungen

Bereits während der Erstellung des Lastenhefts wurde beschlossen, das Funktionsmuster als Client-Server-Anwendung zu implementieren. Als Realisierung dieser Entscheidung wurde beschlossen, das Funktionsmuster in Form eines Web-Service zu implementieren. Dies hat den Vorteil, dass ein Zugriff auf die Benutzeroberfläche von beliebigen Netzwerkteilnehmern möglich ist. Es ist jedoch auch eine lokale Installation möglich. Im Rahmen des Grobentwurfs der Systemarchitektur wurden die dazu notwendigen Technologien festgelegt. Die Kommunikation zwischen dem Client und dem Webservice erfolgt mittels SOAP-Nachrichten, die auf Basis einer WSDL-Spezifikation [4], ausgetauscht werden.

3.1 Vorstellung Web-Service

Die Server-Seite des Funktionsmusters wurde als Web-Service auf Basis der Windows Communication Foundation (WCF) des .Net-Frameworks (https://dotnet.microsoft.com/) von Microsoft implementiert.

Die Schnittstelle des Web-Service bildet die geplanten Funktionen des Funktionsmusters ab und stellt diese dem Client bereit. Abbildung 1 fast den Funktionsumfang des Web-Service zusammen. Mit Hilfe eines Rollenkonzepts kann zwischen den beiden Rollen IT-Security Beauftragter und IT-Security Spezialist unterschieden werden. Der IT-Security Beauftragte führt die täglichen Arbeiten mit dem Funktionsmuster durch. Dies sind das Bearbeiten von Projekten, der Import von Anlageninformationen (AP 8) sowie die Durchführung einer IT-Sicherheitsanalyse (AP 10) und das Generieren der Dokumentation einer Analyse. Die Dokumentation kann für zwei Adressatenkreise erstellt werden. Für den IT-Security Beauftragten, der die ermittelten Schutzmaßnahmen umsetzen muss und das Management, das einen Überblick über die durchgeführte Analyse und deren Konsequenzen benötigt. Der Import von Anlageninformationen erfolgt auf Basis der Beschreibungssprache AutomationML [5], welches zur Laufzeit mittels des „Computer Aided Engineering eXchange“-Datenmodells (CAEX) [6] abgebildet wird. CAEX ist ein generisches Datenmodell, mit dessen Hilfe ein systemübergreifender, herstellerunabhängiger Austausch von Anlageninformationen zwischen beliebigen Entitäten möglich wird. AutomationML dient als „eXtended Markup Language“ (XML)-basiertes Abbildungsformat für das Ablegen des CAEX-Datenmodells in einer Datei (für eine genauere Beschreibung siehe AP 7). Die Projektverwaltung ermöglicht es, für beliebige Anlageninformationen eine gesonderte IT-Sicherheitsanalyse durchzuführen. Ein Projekt enthält neben bestimmten Einstellungen für die Analyse und Informationen zu der Anlage und dem Autor auch die Ergebnisse der letzten Analyse.

Neben den Tätigkeiten des IT-Security Beauftragten kann der IT-Security Spezialist zusätzlich noch die Wissensbasis (AP 11) manuell bearbeiten, falls dies notwendig ist. Alternativ kann die Wissensbasis auch über eine Aktualisierungsdatei aktualisiert werden.

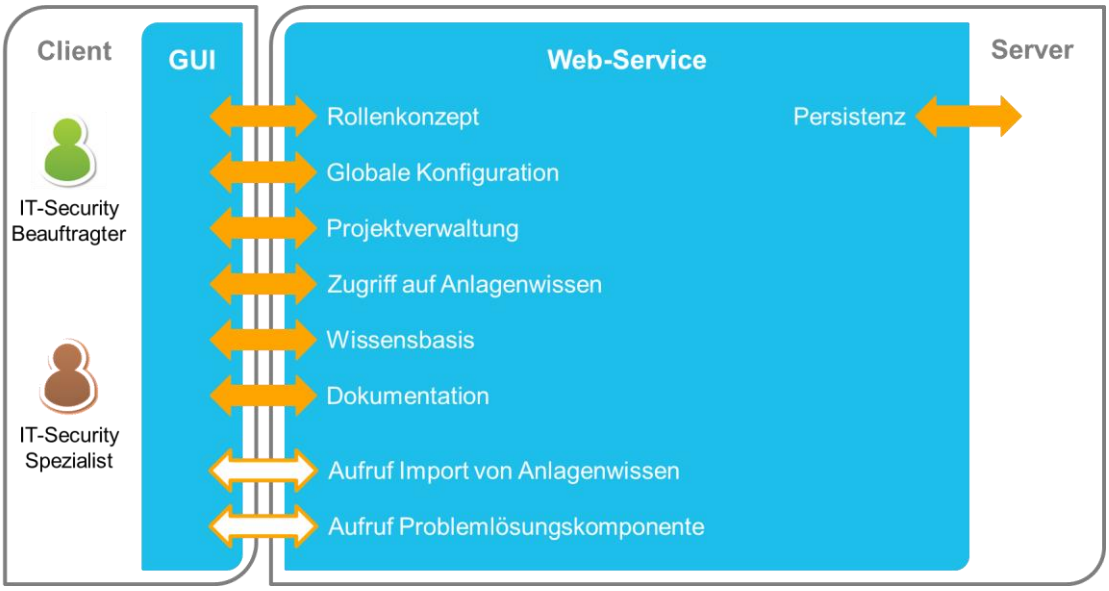

Abbildung 1: Darstellung der Schnittstellen zwischen Web-Service und Benutzeroberfläche (GUI)

Die Wissensbasis enthält jegliches Wissen, das für die Durchführung einer IT-Sicherheitsanalyse notwendig ist. Aufbau und Inhalt der Wissensbasis werden unter AP11 näher behandelt.

3.2 Vorstellung Client / Benutzeroberfläche

Wie in Abbildung 2 zu sehen, besteht die Client-Seite aus der eigentlichen Benutzeroberfläche und einer Middleware. Während die Benutzeroberfläche die Benutzerinteraktion übernimmt, ist die Middleware für die Kommunikation mit dem WCF-Dienst verantwortlich. Der WCF-Dienst stellt die in der Benutzeroberfläche benötigten Daten zur Verfügung.

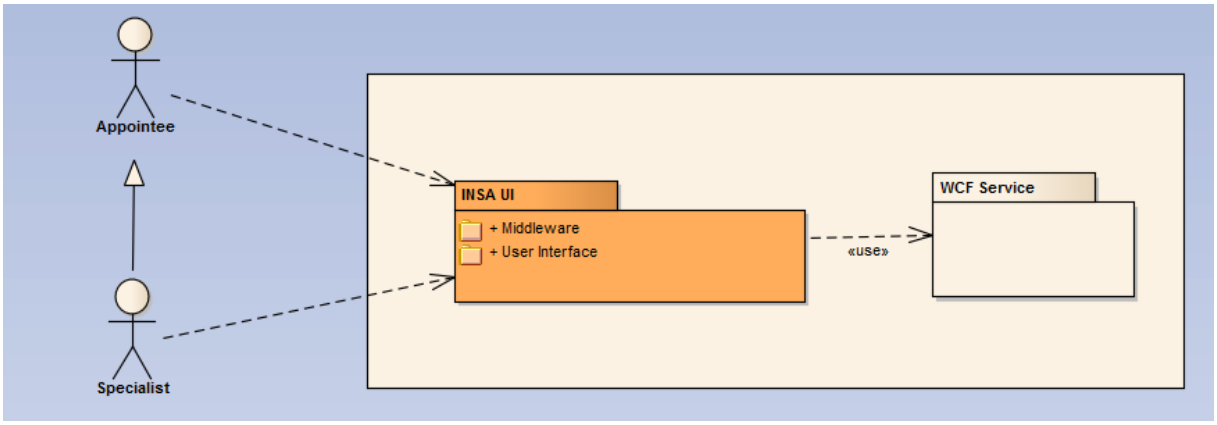

Abbildung 2: Grober Aufbau der Client-Seite

Die Client-Seite des Funktionsmusters wurde mit den nachfolgenden Technologien realisiert (Siehe Abbildung 3):

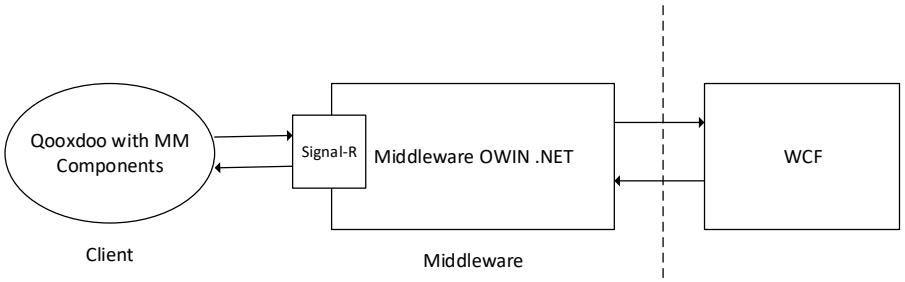

Abbildung 3: Verwendete Technologien der Client-Seite

- Qooxdoo (http://www.qooxdoo.org/):

  Qooxdoo ist ein objektorientiertes JavaScript-Framework. Qooxdoo wurde verwendet um die Benutzerschnittstelle in reinem JavaScript zu entwickeln.

- Signal-R (https://dotnet.microsoft.com/apps/aspnet/signalr):

  SignalR kann verwendet werden, um jede Art von "real-time" Web-Funktionalität zu einer .NET-Anwendung hinzuzufügen. In diesem Szenario, siehe Abbildung 3, wird Signal-R für die Kommunikation zwischen dem Client und der Middleware verwendet. Funktionsaufrufe auf der Middleware können direkt aus JavaScript-Code ausgeführt werden. Die Daten selbst werden im JSON-Format übertragen.

- Middleware OWIN .NET (http://owin.org/):

  Die Middleware wird verwendet um die Abhängigkeiten zwischen Client und WCF-Dienst zu verringern.

3.3 Vorstellung logischer Ablauf der Hauptfunktionen

Im Folgenden sollen die logischen Abläufe bei der Nutzung der Hauptfunktionen beschrieben werden. Punkt 1 in Abbildung 4 zeigt die Anmeldung einer Rolle an dem Web-Service. Dazu muss die bedienende Person sich über die Benutzeroberfläche unter Angabe der Rolle und eines Passworts anmelden. Danach stehen alle der Rolle erlaubten Funktionen zur Verfügung. Eine davon ist das Anlegen(/Laden) eines neuen Projekts (siehe Punkt 2 in Abbildung 4). Intern erstellt die Projektverwaltung ein neues Projekt mit den durch die bedienende Person eingegebenen Informationen wie Name oder die Anlagenbezeichnung (siehe Punkt 3). Gleichzeitig oder im Nachgang können Anlageninformationen in das Projekt geladen werden. Diese werden in dem erzeugten Projekt abgelegt (siehe Punkt 6). Während des Imports kann das wissensbasierte System die Anlageninformationen um weitere Informationen anreichern (siehe Punkt 5, Inhalte in AP 8). Für jedes Projekt wird eine eigene Datei angelegt, die bei jeder Änderung aktualisiert wird. Der Import von Anlageninformationen erfolgt über die Asset-Ansicht der Benutzeroberfläche.

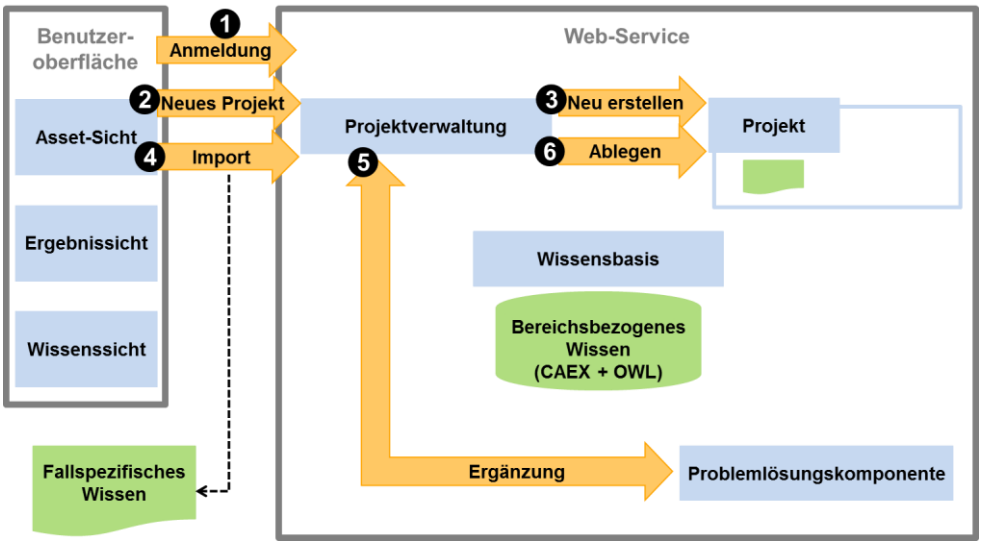

Abbildung 4: Beispiel für das Anlegen eines Projekts und den Import von Anlageninformationen

Ist ein Projekt geladen und liegen Anlageninformationen in dem Projekt vor, kann eine IT-Sicherheitsanalyse auf Basis der Anlageninformationen und der Wissensbasis durchgeführt werden. Die Analyse wird über die Asset-Sicht gestartet (siehe Punkt 7 in Abbildung 5). Die eigentliche Analyse wird durch die Problemlösungskomponente des wissensbasierten Systems durchgeführt. Diese wendet das in der Wissensbasis (bereichsbezogenes Wissen) enthaltene Wissen auf die in einem Projekt enthaltenen Anlageninformationen (fallspezifisches Wissen) an (siehe Punkt 8), um mögliche Schutzmaßnahmen für die/den analysierte(n) Anlagen(-teil) abzuleiten. Das Ergebnis der Problemlösungskomponente wird auch als Lösung bezeichnet. Die Lösung einer Analyse wird dem zugehörigen Projekt hinzugefügt (siehe Punkt 9). Die bedienende Person bekommt daraufhin die Ergebnissicht der Analyse angezeigt, in der die Lösung graphisch aufbereitet dargestellt wird (siehe Punkt 10). Zusätzlich kann die bedienende Person an dieser Stelle auch die Dokumentation in PDF-Form generieren lassen.

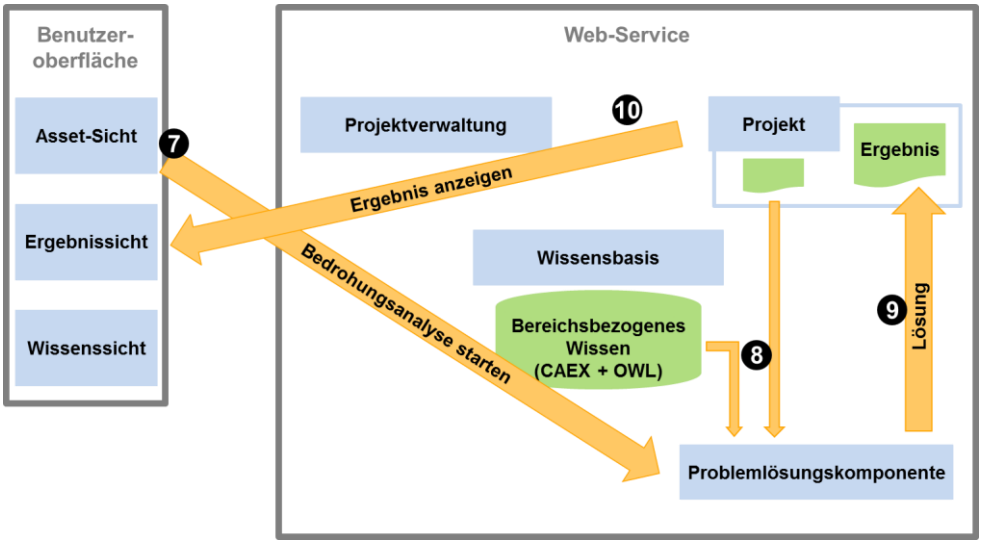

Abbildung 5: Beispiel für die Vorgehensweise bei der Durchführung einer IT-Sicherheitsanalyse

Ist die bedienende Person mit der Rolle IT-Security Spezialist angemeldet, besteht für Sie zusätzlich die Möglichkeit, die Wissensbasis zu bearbeiten (siehe AP 11). Dort können Bedrohungen,

Schutzmaßnahmen und die Regeln bearbeitet werden (siehe Punkt 11 von Abbildung 6). Für die Bearbeitung der Wissensbasis stellt die Benutzeroberfläche eine eigene Wissenssicht bereit, die das Wissen der Wissensbasis übersichtlich darstellt und die bedienende Person bei der Bearbeitung der Wissensbasis unterstützt.

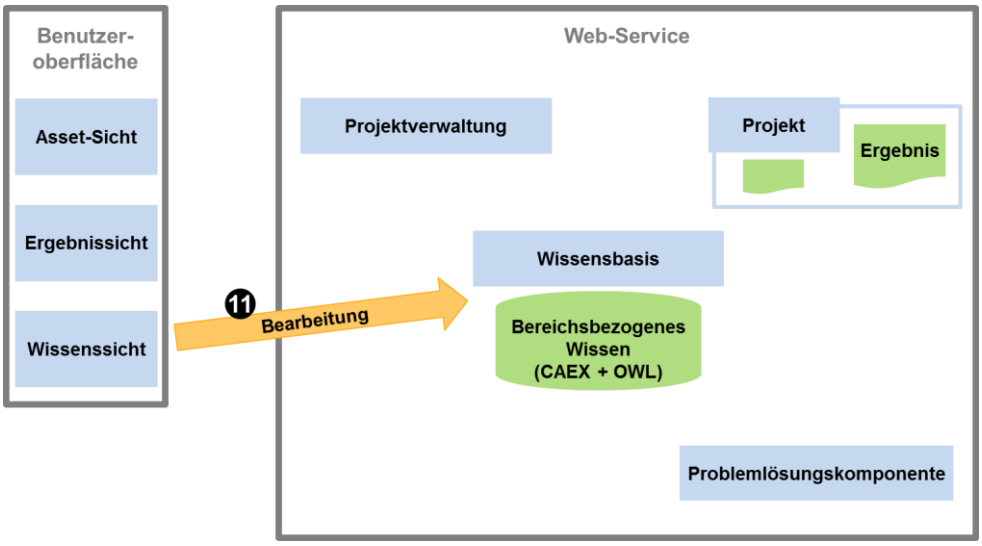

Abbildung 6: Beispiel für die Vorgehensweise für die Bearbeitung des IT-Sicherheitswissens

4 Erfassen der Schwachstellen und Bedrohungen (AP 5)

Maßgeblich für eine systematische Erfassung und Bestimmung relevanter Bedrohungen ist die Bestimmung bzw. Festlegung des zu schützenden Objektes. Dieses wird im Weiteren als „Asset“ bezeichnet. Eine kurze Analyse zeigte, dass sich die betrachteten Assets prinzipiell nur geringfügig von klassischen IT-Systemen unterscheiden. Dieses Resultat bildet eine wichtige Grundlage für die weiteren Arbeitsschritte.

Randbereiche, wie sie nach gängigen IT-Sicherheitsstandards ebenfalls betrachtet werden – hierzu gehören infrastrukturelle sowie organisatorische Aspekte – werden in der hier dargestellten Analyse nur rudimentär betrachtet. Eine weitere, wesentliche Unterscheidung zur klassischen IT bildet die nachvollziehbare, starke Fokussierung auf die Verfügbarkeit einer automatisierungstechnischen Anlage – gegenüber der Vertraulichkeit als wesentliches Schutzziel im Bereich der klassischen IT. Gleichwohl sollte die Vertraulichkeit nicht vollständig vernachlässigt werden, da in automatisierungstechnischen Anlagen ggf. auch Firmengeheimnisse in Gestalt eines spezifischen, entwickelten Fertigungsprozesses gespeichert sein können.

Ausgehend von diesen Grundannahmen wurden die zu betrachtenden Assets zunächst systematisch bezüglich ihrer Eigenschaften und wie sich diese darstellen und bezeichnen lassen, untersucht.

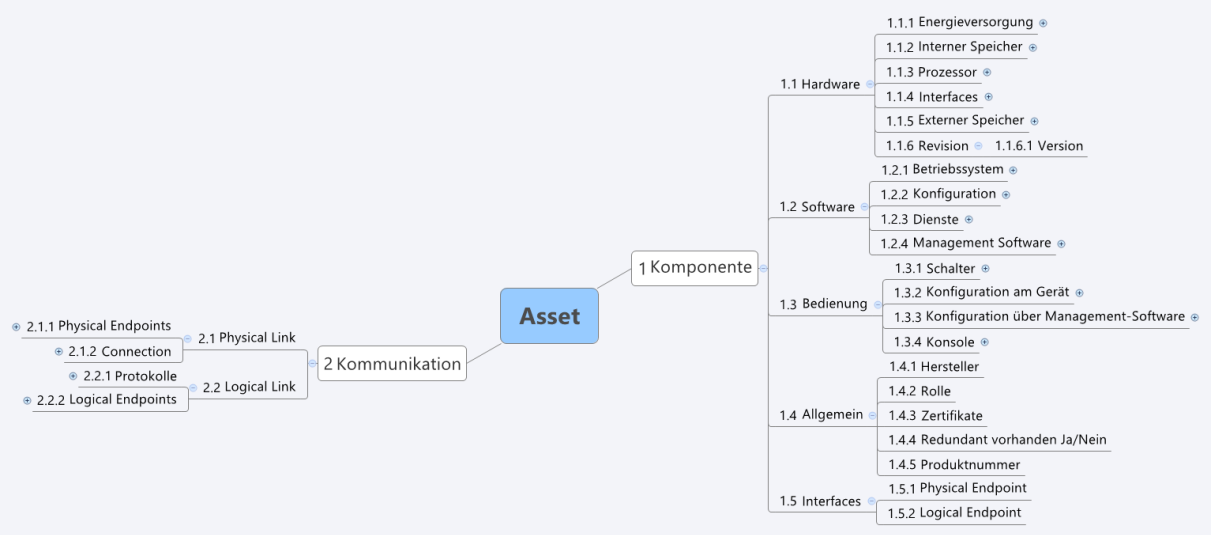

Abbildung 7: Strukturierung der wesentlichen Eigenschaften eines Assets

Die Abbildung 7 zeigt die systematische Darstellung aller wesentlichen Eigenschaften eines Assets. Diese Eigenschaften werden im Weiteren für die Analyse bzw. Bestimmung möglicher Bedrohungen herangezogen.

Die Bedrohungen selbst wurden primär auf der Grundlage etablierter Kataloge und Regelwerke erstellt. Hierbei wurden u.a. die folgenden Quellen herangezogen:

- BSI Grundschutz-Kataloge (spezifische und generische Gefährdungskataloge),
- BSI Sicherheitshandbuch sowie
- Literatur aus dem Bereich IT-Sicherheit für ICS.
- Betreuung mehrerer Bachelorarbeiten im Themenfeld IT-Sicherheit automatisierungstechnischer Anlagen

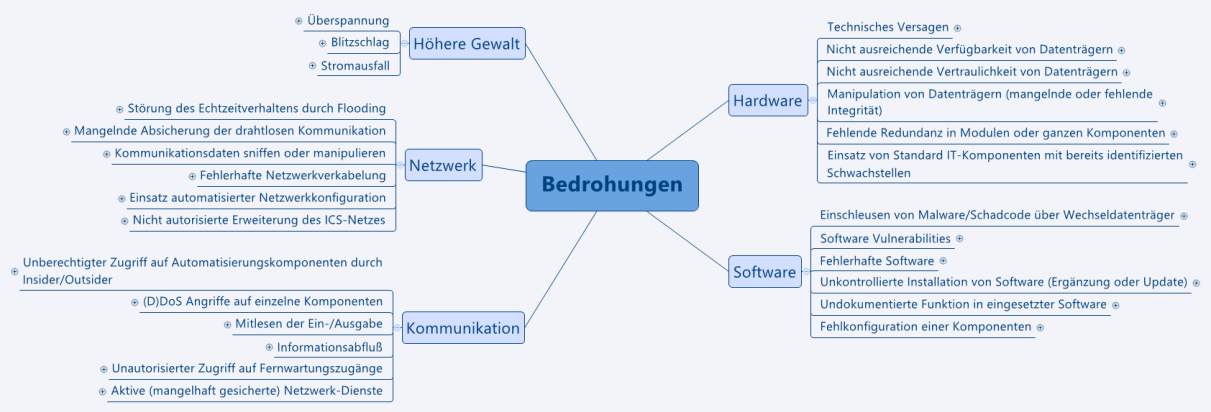

Abbildung 8: Strukturierung der Bedrohungen.

Die Abbildung 8 zeigt die Struktur, wie die ermittelten Bedrohungen insgesamt klassifiziert und dargestellt worden sind.

Eine Reihe dieser Bedrohungen lässt sich nahezu direkt aus konkreten einzelnen Eigenschaften eines Assets oder aus deren Kombination, wie sie in der Abbildung 7 dargestellt sind, ableiten. Andere Bedrohungen sind nur indirekt aus einzelnen Eigenschaften direkt abzuleiten. Hierzu gehört die gesamte Gruppe zur Kommunikation.

Die Bestimmung der möglichen Bedrohungen einer automatisierungstechnischen Anlage ist aber nur eine Seite der Sicherheitsanalyse. Die andere Seite bildet die Festlegung bzw. Bestimmung der notwendigen Sicherheitsmaßnahmen. Diese wurden ausgehend von den ermittelten, relevanten Bedrohungen unter Anwendung der folgenden Randbedingungen bestimmt:

- Ausgangspunkt sind „klassische“ IT-Sicherheitsmaßnahmen.
- Einzelne Maßnahmen werden bezüglich der speziellen Einsatzbedingungen einer industriellen Steuerungsanlage modifiziert oder ergänzt.
- Die ermittelten Maßnahmen werden nach klassischen Kategorien gruppiert.

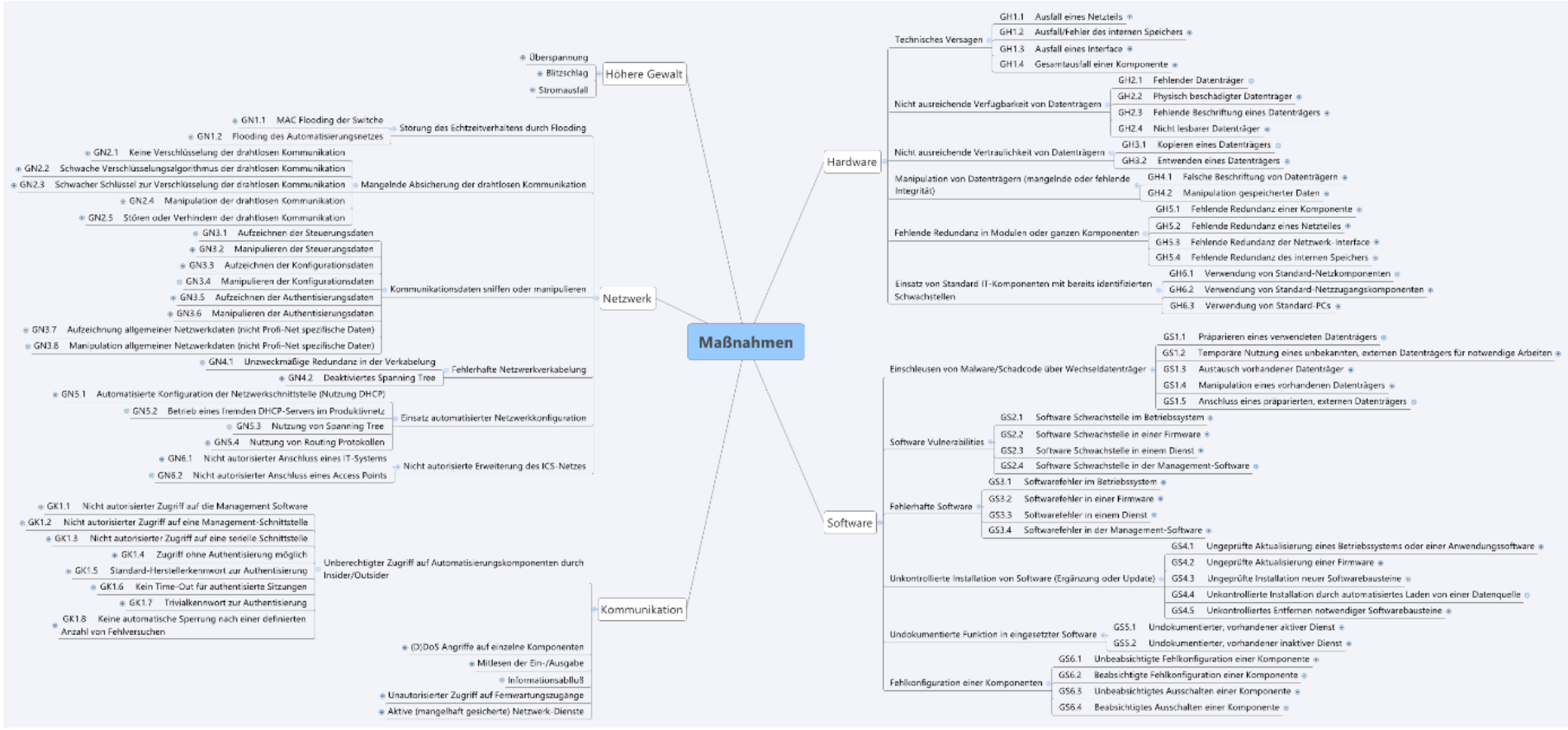

Abbildung 9: Übersicht der ermittelten Maßnahmen.

Abbildung 9 zeigt nicht direkt die aus den Bedrohungen abgeleiteten Maßnahmen, sondern zeigt noch einmal detailliert alle Bedrohungen, wie sie schon in der Abbildung 8 gruppiert gezeigt werden. Jede einzelne Bedrohung ist darin mit absichernden Maßnahmen verknüpft.

Gruppiert wurden die so erhaltene Menge der Maßnahmen in die folgenden Kategorien:

- Hard- und Software (16 Maßnahmen)
- Infrastruktur (6 Maßnahmen)
- Kommunikation (17 Maßnahmen)
- Netzwerk (10 Maßnahmen)
- Organisation (30 Maßnahmen)

Alle ermittelten Maßnahmen wurden ergänzend einer Einstufung nach zwei Kriterien unterzogen:

Aufwand	Was kostet die Umsetzung der Maßnahme?
Wirksamkeit	Welcher Nutzen kann mit der Umsetzung dieser Maßnahme realisiert werden?

Der Aufwand wurde in drei Stufen betrachtet:

Tabelle 1: Einstufung des Implementierungsaufwands für Schutzmaßnahmen

Aufwand	Erläuterung
Gering	Maßnahmen, die mit wenigen Handgriffen erledigt werden können bspw. unter Zuhilfenahme des entsprechenden Handbuches des jeweiligen Assets oder der Software. Dazu gehören kleinere Änderungen in den Konfigurationen einzelner Assets, Software-Anwendungen oder Betriebssystemen.
Mittel	Maßnahmen, die einen moderaten Aufwand zur Umsetzung benötigen. Dazu gehören Installationen und Konfigurationen von (separaten) Software-Anwendungen, Änderungen an der Hardware oder das Verfassen und Umsetzen von Richtlinien.
Hoch	Maßnahmen, die umfangreichen Aufwand zur Umsetzung benötigen, ggf. unter Zuhilfenahme von (externen) spezialisierten Fachkräften. Dazu gehören Neukonfigurationen des Netzes, Umstrukturierung des Netzes, Repositionierung der Assets oder Hardware-Anschaffungen und deren Implementation.

Die Wirksamkeit wurde ebenfalls in drei Stufen betrachtet, die allerdings jeweils noch einmal in zwei Untergruppen aufgeteilt worden sind:

Tabelle 2: Wirksamkeitsstufen von Schutzmaßnahmen

Wirksamkeit	Erläuterung
Punktuell einfach	Die Maßnahme schützt ein einzelnes Asset sowie ggf. die daran direkt angeschlossenen Sensoren oder Aktoren. Sie kann mit Fachwissen umgangen werden.

Punktuell anspruchsvoll	Die Maßnahme schützt ein einzelnes Asset sowie ggf. die daran direkt angeschlossenen Sensoren oder Aktoren. Sie kann nur mit detailliertem Fachwissen sowie ggf. notwendigen Werkzeugen (Tools) umgangen werden.
Übergreifend einfach	Die Maßnahme schützt eine Gruppe von Assets mit allen damit verbundenen weiteren Assets sowie Sensoren und Aktoren. Sie kann mit Fachwissen umgangen werden.
Übergreifend anspruchsvoll	Die Maßnahme schützt eine Gruppe von Assets mit allen damit verbundenen weiteren Assets sowie Sensoren und Aktoren. Sie kann nur mit detailliertem Fachwissen sowie ggf. notwendigen Werkzeugen (Tools) umgangen werden.
Hochwertig	Die Überwindung der Maßnahmen bedarf tiefgreifender technischer Kenntnisse und Fähigkeiten und setzt einen sehr hohen Aufwand seitens des Angreifers voraus.
Best Solution	Die Maßnahmenzusammenstellung deckt nahezu alle derzeit bekannten Angriffsmuster ab. Ein erfolgreicher Angriff kann nur durch noch unbekannte Schwachstellen und Sicherheitslücken erfolgen.

Mit diesen beiden Parametern sind Anwender in der Lage, Maßnahmen nach ihren konkreten und individuellen Bedürfnissen auszuwählen, weil eine Bedrohungs- und Risikoanalyse natürlich immer im Kontext des jeweiligen Unternehmens stattfindet, d.h. eine Bewertung einer potentiellen Maßnahme nur dann möglich ist, wenn einerseits der zur Umsetzung notwendige Aufwand (monetär und/oder personell) sowie andererseits der zu erwartende Schutz in Form einer grundsätzlichen Einstufung beschrieben bzw. klassifiziert sind.

Die folgende Tabelle zeigt die Einstufung der im Rahmen der Bedrohungsanalyse ermittelten Maßnahmen.

Tabelle 3: Einstufung der im Rahmen der Bedrohungsanalyse ermittelten Schutzmaßnahmen

	Aufwand		
	Gering	**Mittel**	**Hoch**
Punktuell einfach	9	3	1
Punktuell anspruchsvoll	3	3	2
Übergreifend einfach	(1)	11	7
Übergreifend anspruchsvoll	8	9	7
Hochwertig	2	8	2
Best Solution	(1)	(1)	2

Eine weitere, alternative Möglichkeit, Maßnahmen in ihrem Aufwand und damit auch in ihrer Wirksamkeit zu differenzieren wurde durch die Einführung von Protection-Level (PL) geschaffen [7]. Hierbei wurden beispielhaft einzelne Maßnahmen im notwendigen Aufwand und damit letztlich im Grade ihrer Wirksamkeit auf vier Stufen differenziert.

Um die PL abzuleiten, werden zunächst zwei einzelne Parameter näher betrachtet und jeweils in vier Klassen eingeteilt:

- Sicherheitsfunktionalität (Security Level)

- Sicherheitsprozess (Maturity Level)

Tabelle 4: Einstufung von Sicherheitsfunktionalitäten

Level	Sicherheitsfunktionalität
SL 1	Fähigkeit zum Schutz gegen unbeabsichtigte Bedrohungen.
SL 2	Fähigkeit zum Schutz gegen beabsichtigte Bedrohungen, die durch geringe Ressourcen, generische Kenntnisse und niedriger Motivation entstehen.
SL 3	Fähigkeit zum Schutz gegen beabsichtigte Bedrohungen, die durch mäßigen Ressourcen, spezifischen Kenntnisse und mäßiger Motivation entstehen.
SL 4	Fähigkeit zum Schutz gegen beabsichtigte Bedrohungen, die durch umfassende Ressourcen, spezifischen Kenntnisse und hoher Motivation entstehen

Tabelle 5: Einstufung von Sicherheitsprozessen

Level	Sicherheitsprozess
ML 1	Initialer, unvorhersehbarer Prozess, schwach kontrolliert und reaktiv.
ML 2	Gesteuerter, beschriebener Prozess, reaktiv.
ML 3	Definierter und beschriebener Prozess, proaktive Bereitstellung.
ML 4	Optimierter und messbarer Prozess, kontrolliert und kontinuierlich verbessert.

Die beiden Klassifizierungen werden durch die folgende Tabelle miteinander verknüpft.

Tabelle 6: Ermittlung des Protection-Levels anhand der SL und ML

		Security Level			
		SL 1	**SL 2**	**SL 3**	**SL 4**
Maturity Level	**ML 1**	PL 1	PL 1	PL 1	PL 1
	ML 2	PL 1	PL 2	PL 2	PL 2
	ML 3	PL 1	PL 2	PL 3	PL 4
	ML 4	PL 1	PL 2	PL 3	PL 4

Um dieser Einteilung gerecht zu werden, wurden einzelne Maßnahmen beispielhaft in ihren Beschreibungen derart modifiziert, dass eine Abstufung bezüglich ML einerseits sowie bezüglich SL andererseits möglich wird.

5 Erstellung Funktionsmuster entsprechend Grobentwurf der Systemarchitektur (AP 7)

Auf Basis der Vorarbeiten aus AP 4 wurde die grundlegende Architektur des Funktionsmusters implementiert. Dies beinhaltete den grundsätzlichen Aufbau des Funktionsmusters mit den notwendigen Schnittstellen für die Entwicklung der Inhalte aus den nachfolgenden Arbeitspaketen.

6 Import der Projektierungsdaten aus Engineering-Werkzeug (AP 8)

Wie bereits zuvor beschrieben, wurde für den Import der Anlagendaten das Datenaustauschformat AutomationML gewählt. Der Kern von AutomationML stellt hierbei eine Implementierung des CAEX-Schemas nach IEC 62424 [6] dar. AutomationML bietet als offenes, anpassbares und standardisiertes Datenformat die Möglichkeit, die benötigten Informationen über die Grenzen verschiedener Engineering Werkzeuge hinweg auszutauschen. Aktuell planen mehrere Hersteller von Engineering Lösungen den Einsatz von AutomationML als Datenaustauschformat bzw. realisieren dies bereits [8]. Da diese Funktionen sich jedoch bei den Werkzeugherstellern noch in der Entwicklung befinden, konnte der direkte Export aus Engineering-Werkzeugen bisher nicht wie geplant vorgenommen werden. Jedoch wurde bei der Definition des Austauschformats diese Erweiterungsmöglichkeit berücksichtigt, so dass eine spätere Anpassung möglich ist.

Die Wahl von AutomationML als Importformat wurde durch den Fakt bestärkt, dass der Aufbau des zugrunde liegenden CAEX-Metamodels einem wissensbasierten System entgegenkommt. CAEX bietet die Möglichkeit, sowohl allgemeines Wissen über möglichen Automatisierungskomponenten, ihren Aufbau, Schnittstellen und Rollen als bereichsbezogenes Wissen als auch den expliziten Aufbau der betrachteten automatisierungstechnischen Anlage als fallspezifisches Wissen in einer Datei zu bündeln.

Für die IT-Sicherheits-relevanten Informationen in AutomationML wurde zunächst auf der Basis der in AP 3 ermittelten Beispielanlagenstruktur ein Schema zur Abbildung entwickelt. Auf Basis dieses Schemas wurde die Beispielstruktur in einem Ausschnitt mit möglichen Automatisierungskomponenten hinterlegt und diese in AutomationML nachgebildet. Der Anlagenausschnitt sowie die Darstellung in AML sind in Abbildung 6 dargestellt.

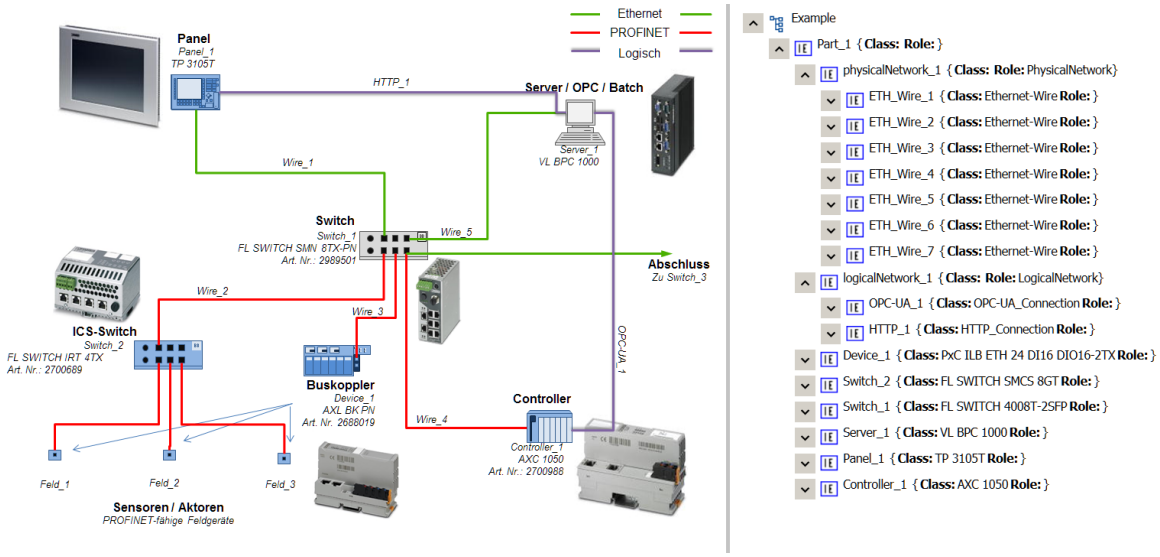

Abbildung 6: Anlagenausschnitt und Repräsentation in AML

Es wurde davon ausgegangen, dass die importierten Daten in der AutomationML-Datei nicht zwingend alle Informationen abbilden, die als Grundlage einer IT-Sicherheitsanalyse benötigt werden. Dies ergibt sich hauptsächlich dadurch, dass hier zum Teil sehr spezielle Information, z.B. über verwendete Softwareversionen, erforderlich ist, die in anderen Engineering-Werkzeugen nicht abgebildet wird. Auf der Grundlage des Aufbaus der AutomationML-Datei wurde eine Möglichkeit zur automatischen Ergänzung der fehlenden Informationen entwickelt und implementiert. Durch die so hinzugewonnenen Informationen konnten die Ergebnisse der anschließenden IT-Sicherheitsanalyse verbessert werden. Eine mögliche Ergänzung von Wissen anhand eines Industrie-Servers von PxC ist in Abbildung 7 dargestellt.

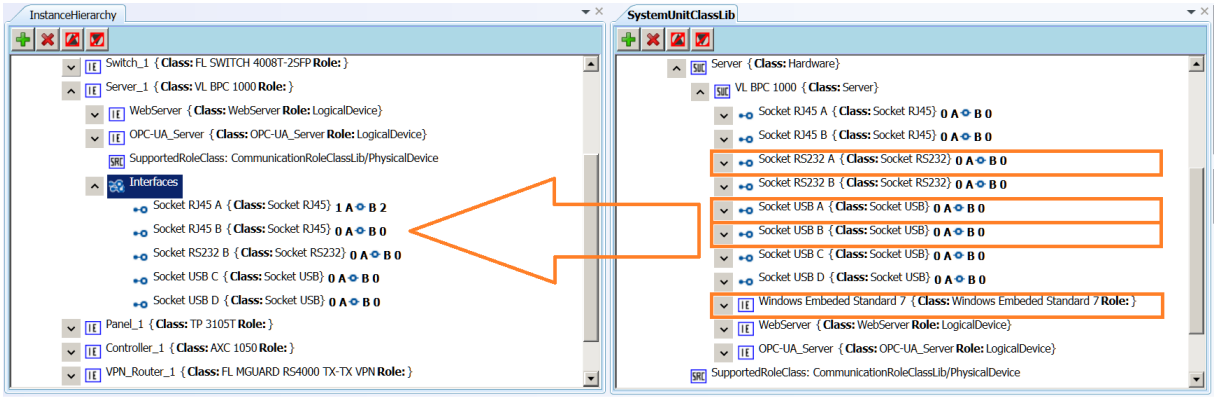

Abbildung 7: Wissensergänzung während des Imports

Es ist davon auszugehen, dass die betrachtete automatisierungstechnische Anlage im Verlauf der Betriebszeit Änderungen unterworfen ist. Diese Änderungen müssen in die IT-Sicherheitsanalysen Eingang finden. Um dies abzubilden, wurde die Möglichkeit geschaffen, in einem bestehenden Projekt einen Re-Import der aktualisierten Anlageninformationen durchzuführen. Da die bereits im Projekt vorliegenden Anlageninformationen durch die genannten automatischen Ergänzungen und durch manuelle Eingaben erweitert wurden, würde ein Ersetzen der vorliegenden Informationen durch den neuen Import möglicherweise zu Informationsverlust führen und unnötige Mehrarbeit durch die anwendende Person nach sich ziehen. Um diesem zu begegnen, wurde auf Basis der AutomationML Engine ein Delta-Vergleich zur Ermittlung von Unterschieden zwischen dem vorhandenen und dem neu importierten Dokument entwickelt. Auf der Grundlage dieser ermittelten Unterschiede kann die anwendende Person für jedes einzelne abweichende Element entscheiden, welcher Versionsstand übernommen werden soll, und wird somit von unnötiger Mehrarbeit entlastet. Eine Verwendung dieser Funktionalität auch für andere softwaregestützte Lösungen ist möglich.

7 Problemlösungskomponente (AP 10)

Um eine Durchführung einer automatisierten wissensbasierten IT-Sicherheitsanalyse zu ermöglichen, musste zunächst eine Möglichkeit der Verbindung (a) der Anlageninformationen und des anlagenspezifischen bereichsbezogenen Wissens in der AutomationML-Datei und (b) des IT-Security-Wissens in Form einer Ontologie (Beschreibung siehe AP 11) geschaffen werden. Es konnte ein neues Konzept zur Verbindung von Engineering Information mit Ontologie basiertem Wissen entwickelt, veröffentlicht und im Rahmen des vorliegenden Projektes implementiert werden. Das hierbei entwickelte Konzept ist nicht nur für die betrachteten Security Lösungen anwendbar, sondern kann auch für die wissensbasierte Realisierung weiterer Engineering-Aufgaben Anwendung finden.

Auf der Basis dieser Anbindung der Engineering-Informationen an das IT-Security-Wissen konnte eine wissensbasierte Erstellung von IT-Security-Konzepten durch Anwendung des erstellten Regelwerks vorgenommen werden. Der Umfang des in der Beispielanalyse erstellten Regelwerks und die daraus abzuleitenden Umfänge für ein vollumfängliches Regelwerk zeigten jedoch, dass im Rahmen der Problemlösung Maßnahmen zur Reduktion der Laufzeit einer IT-Sicherheitsanalyse notwendig wurden. Um die Laufzeit der Analysen zu reduzieren, war es notwendig, die Wissensbasis und damit das Verfahren zur Lösungsfindung zu strukturieren. Hierzu wurden die erstellten Regeln anhand eines aus der aktuellen Normenlage abgeleiteten generischen Vorgehensmodells in Regelklassen eingeteilt. Das generische Vorgehensmodell ist in Abbildung 8 dargestellt.

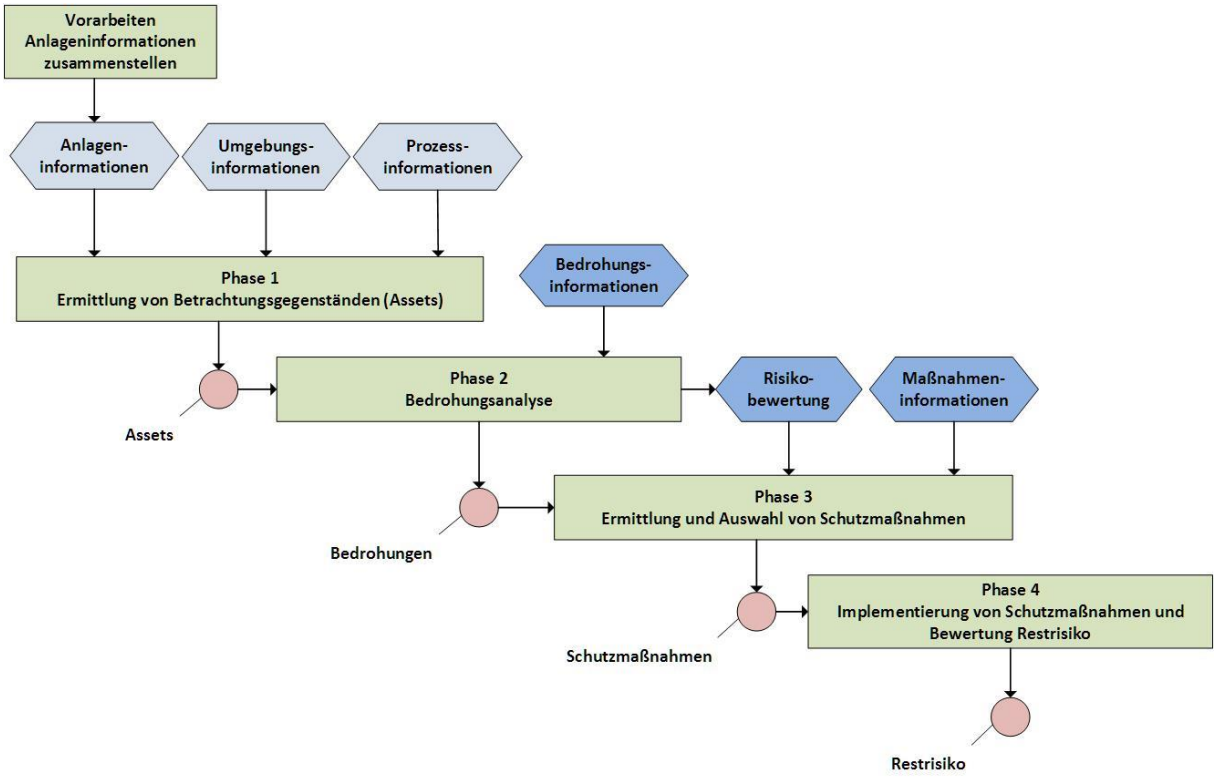

Abbildung 8: Generisches Vorgehensmodell einer IT-Security Analyse

Neben der logischen Reihung können die Regeln anhand unterschiedlicher Kriterien, wie z.B. Hersteller, Produktkategorie, etc., in verschiedene Gruppen eingeteilt werden. Die Regeln einer Gruppe müssen hier nur Anwendung finden, wenn mindestens ein Element der betrachteten Anlage von diesen betroffen ist. Befindet sich also zum Beispiel keine Komponente des Herstellers X in einer

Anlage, so können alle Regeln, die nur für diesen Hersteller zutreffen, von der Lösungsfindung ausgeschlossen werden.

Durch die vorgestellten Strukturierungen der Lösungsfindung konnte die Laufzeit mit den aktuellen Testfällen um 30-50% reduziert werden. Hierbei fand aufgrund des beispielbezogenen Inhalts der Wissensbasis hauptsächlich das logische Strukturierungskriterium der Regelklassen Anwendung. Regelgruppen greifen nach dem aktuellen Stand der Arbeiten erst bei der Erstellung einer vollumfänglichen Wissensbasis. Die Abbildung von Regelklassen und Regelgruppen wird in AP 11 näher erläutert.

8 Erstellung der Wissensbasis (Zusammenfassung des gesamten Strukturwissens) (AP 11)

Im Rahmen des APs erfolgte die Spezifikation für die Abbildung des benötigten Wissens für eine IT-Sicherheitsanalyse in ein geeignetes Datenschema. Des Weiteren erfolgte die Formalisierung des Wissens (auf Basis von AP 5) sowie die Implementierung notwendiger Software-Komponenten für den Zugriff auf und die Verwaltung des Wissens.

8.1 Aufbau der Wissensbasis

Der Einsatz eines regelbasierten Systems bietet den Vorteil, dass die Wissensbasis flexibel und schnell an die ständig wechselnde Bedrohungslage von automatisierungstechnischen Anlagen angepasst und nach Bedarf erweitert werden kann. Somit bietet das Funktionsmuster die Möglichkeit, eine IT-Sicherheitsanalyse jeweils mit dem aktuellen Stand des Wissens durchzuführen.

Die Wissensbasis enthält alle Informationen bzw. das gesamte Wissen, das für die Durchführung der IT-Sicherheitsanalyse benötigt wird. Dieses unterteilt sich in die Bereiche des bereichsbezogenen Wissens (IT-Sicherheitswissen) und des fallspezifischen Wissens (Anlagenwissen). Der Aufbau der Wissensbasis wird in Abbildung 10 dargestellt, welche auch die Verteilung des Wissens innerhalb des Funktionsmusters abbildet.

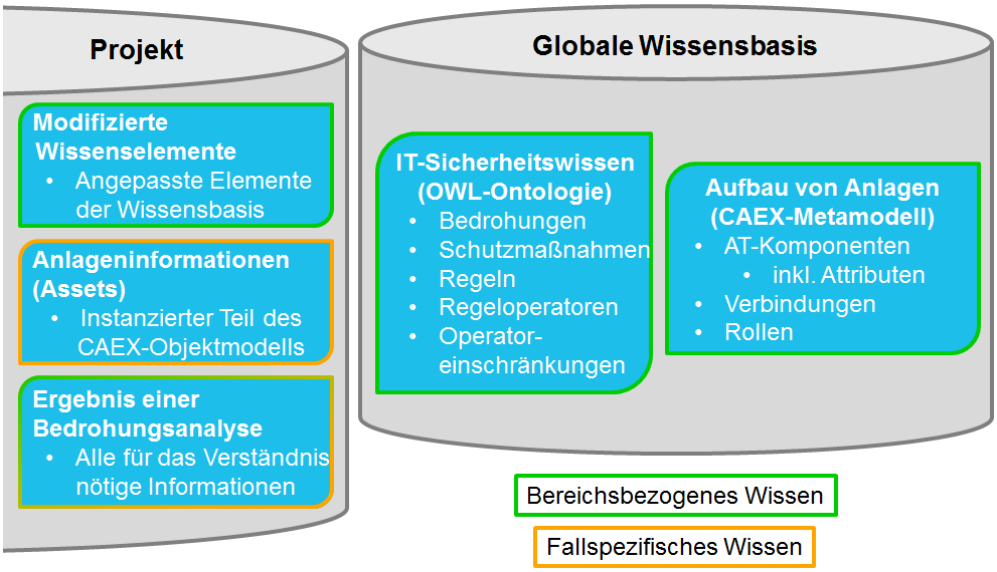

Abbildung 10: Verteilung des Wissens im Funktionsmuster

Das fallspezifische Wissen umfasst Informationen über eine spezifische zu analysierende Anlage im AutomationML-Format bzw. CAEX (siehe AP 10). Dieses Wissen ist ausschließlich in den Projekten gespeichert. Das bereichsbezogene Wissen enthält jedoch generelles Wissen über automatisierungstechnische Anlagen im AutomationML-Format bzw. CAEX (siehe AP 10). Das restliche bereichsbezogene Wissen (außer den Regeln) wird mit Hilfe der Web Ontology Language (OWL) [9] in Form einer Ontologie abgebildet. Bestandteile wie Bedrohungen und Schutzmaßnahmen werden dabei als OWL-Individuals und Ihre Eigenschaften als OWL-Properties dargestellt. Die Regeln werden mit Hilfe der Semantic Web Rule Language (SWRL) [10] abgebildet. Zusätzlich gibt es noch Regeloperatoren, die Teile der Regeln sind. Diese wurden ebenfalls auf Basis von OWL-Properties entworfen. Da das Wissen über Anlagen jedoch nicht in OWL vorliegt, verweisen manche der

Regeloperatoren auf bestimmte Stellen des CAEX-Modells. Diese sind Teil des in AP 10 bereits vorgestellten Konzepts.

Bereichsbezogenes Wissen, das für alle Projekte gilt, wird in der globalen Wissensbasis vorgehalten. Projekt-spezifisches bereichsbezogenes Wissen wird direkt in den Projekten vorgehalten. Die Projekte enthalten zusätzlich die Ergebnisse einer IT-Sicherheitsanalyse, welche Teile sowohl aus dem fallspezifischen Wissen als auch aus dem bereichsbezogenen Wissen enthalten, die für das Verständnis des Ergebnisses benötigt werden.

Abbildung 11 gibt einen Überblick über den Umfang des bereichsbezogenen Wissens der Wissensbasis. *Thing* ist das oberste Konzept (Basiskonzept von OWL) der Wissensbasis. Eines der Kindkonzepte ist die Enumeration *CaexAttributeConstraint_CAEX_Attribute*. Diese steht stellvertretend für eine beliebige Anzahl von Enumerationen in der Wissensbasis. Diese legen für bestimmte Attribute von Assets fest, welche Werte diese annehmen dürfen. Auf diese Weise kann in der Benutzeroberfläche eine Auswahl von Werten angezeigt werden, um den Benutzer bei der Erstellung einer Regel zu unterstützen und gleichzeitig Fehleingaben zu verhindern. *KnowledgeItem* repräsentiert die eigentlichen Wissenselemente, die bei der Durchführung einer IT-Sicherheitsanalyse benötigt werden. Das Konzept *Item* repräsentiert das Wissen über automatisierungstechnische Anlagen und ist ein Platzhalter für die Verbindung mit dem CAEX-Modell (siehe AP 8 und AP 10).

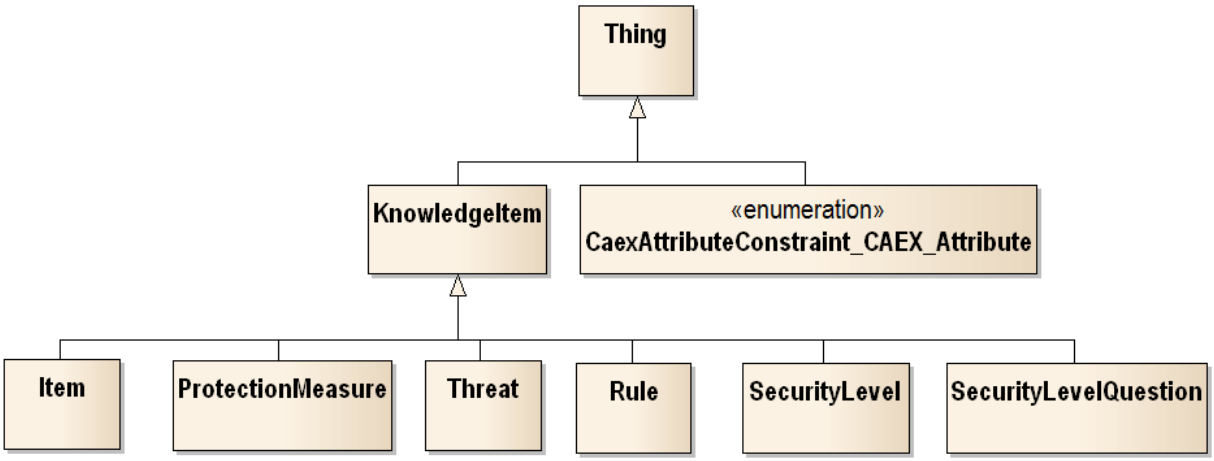

Abbildung 11: Übersicht über den gesamten Aufbau der Wissensbasis

Die Konzepte *SecurityLevel* und *SecurityLevelQuestion* sind Kindelemente von *KnowledgeItem* und bilden verschiedene Schutzniveaus einer Anlage und dazugehörige Einstufungsfragen ab. Nach der Beantwortung der Fragen wird der benötigte Schutzbedarf einer zu analysierenden Anlage automatisch ermittelt. Optional kann die bedienende Person auch selbst den Schutzbedarf festlegen. Die Security Level sind angelehnt an die IEC 62443 [7] und wie folgt festgelegt:

- **Security Level 1:** Schutz gegen zu lockere oder aus Versehen nicht eingehaltenen IT-Security-Vorschriften
- **Security Level 2:** Schutz gegen absichtliche Umgehung der IT-Security-Maßnahmen mit einfachen Maßnahmen, wenigen Ressourcen, geringem Wissen und niedriger Motivation

- **Security Level 3:** Schutz gegen absichtliche Umgehung der IT-Security-Maßnahmen mit fortgeschrittenen Maßnahmen, moderaten Ressourcen, systemspezifischen Wissen und moderater Motivation
- **Security Level 4:** Schutz gegen absichtliche Umgehung der IT-Security-Maßnahmen mit fortgeschrittenen Maßnahmen, erweiterten Ressourcen, systemspezifischen Wissen und hoher Motivation

Das Konzept *Threat* repräsentiert alle Bedrohungen, die für eine automatisierungstechnische Anlage bestehen können, und bildet diese ab. Als Kategorien für Bedrohungen wurden Personal, Hardware, Software, Netzwerk und Kommunikation festgelegt. Des Weiteren enthalten Bedrohungen Informationen über ihre Eintrittswahrscheinlichkeit und ihre Auswirkungen.

Das Konzept *ProtectionMeasure* repräsentiert alle Schutzmaßnahmen, die für eine automatisierungstechnische Anlage umgesetzt werden können, und bildet diese ab. Als Kategorien für Bedrohungen wurden infrastrukturell, organisatorisch, technisch und personell festgelegt. Des Weiteren ist der Aufwand für die Implementierung einer Schutzmaßnahme, ihr aktueller Implementierungsstatus/Reifegrad und ihre Effektivität (Höhe des Schutzniveaus) hinterlegt.

Die Regeln (*Rule*) der Wissensbasis sind ebenfalls ein Kindkonzept von *KnowledgeItem.* Sie verknüpfen Wissen über Items (Assets), Bedrohungen und/oder Schutzmaßnahmen miteinander. Dadurch kann die Problemlösungskomponente bei einem Vergleich mit dem fallspezifischen Wissen (Wissen über konkrete Assets) die benötigten Schutzmaßnahmen für eine spezielle Anlage ableiten. Eine Regel besteht dabei aus mehreren Operatoren, die jeweils durch ein UND verknüpft sind.

Jede Regel kann beliebig vielen Gruppen zugeordnet werden. Die Gruppen können zudem beliebig definiert werden. Dadurch kann die Problemlösungskomponente während der IT-Sicherheitsanalyse schneller erkennen, welche Regeln zu beachten sind und welche im Moment keine Rolle spielen (z. B. bestimmter Hersteller). Dies erhöht die Effektivität einer Analyse.

Regelklassen legen fest, welche Wissenselemente miteinander verknüpft werden dürfen. Jede Regel gehört zu genau einer Regelklasse. Tabelle 7 zeigt die verschiedenen Kombinationsmöglichkeiten.

Tabelle 7: Auflistung aller Regelklassen und ihrer Bedeutung

Regelklasse	Prämisse	Konklusion
RK1	Item.Attribut(e)	Item.Attribut(e)
RK2	Item.Attribut(e)	Bedrohung(en)
RK3	Bedrohung(en)	Bedrohung(en)
RK4	Bedrohung(en) + Item.Attribut(e)	Bedrohung(en)
RK5	Bedrohung(en)	Schutzmaßnahme(n)
RK6	Bedrohung(en) + Item.Attribut(e)	Schutzmaßnahme(n)
RK7	Schutzmaßnahme(n)	Gewählte Schutzmaßnahme(n)
RK8	Schutzmaßnahme(n)+ Bedrohung(en)	Gewählte Schutzmaßnahme(n)
RK9	Schutzmaßnahme(n)+ Item.Attribut(e)	Gewählte Schutzmaßnahme(n)

8.2 Verwaltung der Wissensbasis

Die OWL-Ontologie des Funktionsmusters liegt in Form einer „Ressource Description Framework“(RDF)/XML-Datei [11], welche ein XML-basiertes Speichern der Ontologie in Form von

Tripeln erlaubt, vor. Mit Hilfe einer externen Bibliothek (OwlDotNetApi [12]), welche um SWRL erweitert wurde, kann die Ontologie in das Funktionsmuster geladen (oder aus diesem gespeichert) werden. Die Bibliothek erzeugt zur Laufzeit einen Graphen der Ontologie, der beliebig bearbeitet werden kann. Zusätzlich wird auf Basis des Graphen eine Abstrahierung der Ontologie erzeugt. Diese repräsentiert das Wissen in Form von Objekten. Diese Objekte können dann über die Schnittstelle des Web-Service bereitgestellt werden. Für die Übertragung werden die Objekte in XML serialisiert. Dieser Ablauf wird in Abbildung 12 dargestellt.

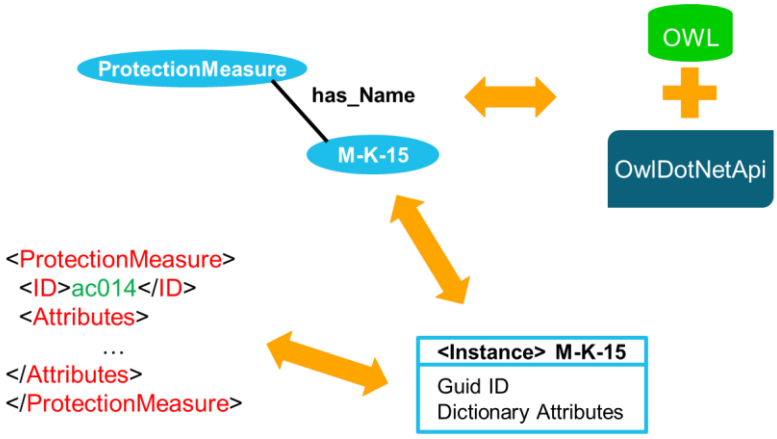

Abbildung 12: Abstraktion der OWL-Ontologie zur Laufzeit des Funktionsmusters

Die Bearbeitung des Wissens erfolgt durch die Veränderung der Objekte über die Benutzeroberfläche. Bevor die Änderung daraufhin in den Graphen übertragen wird, erfolgt zunächst eine Validierung des Wissens, um die Konsistenz der Wissensbasis sicherzustellen.

Die Aktualisierung der Wissensbasis als Ganzes kann durch eine Import-Funktion des Web-Service durchgeführt werden. Dabei kann entweder die Wissensbasis durch eine neue Version ausgetauscht werden oder die bestehende Wissensbasis um weitere Elemente ergänzt werden. Zudem bietet der Web-Service eine Export-Funktion, um die Wissensbasis verteilen zu können.

Durch den Einsatz von OWL und SWRL (als Teil des Semantic Web) können die Wissensbasis und Ihre Konzepte auch auf andere Probleme übertragen werden.

Das Konzept für die Wissensbasis sowie die Inhalte wurden durch die Projektpartner durch Reviews überprüft. Des Weiteren wurde in AP 16 eine manuelle Überprüfung der Korrektheit und Vollständigkeit der Wissensbasis auf Basis der manuell durchgeführten IT-Sicherheitsanalyse (AP 5) durchgeführt.

9 Benutzeroberfläche (AP 12)

Zu Beginn von AP 12 fand zunächst die Konzeption einer nutzergeführten IT-Sicherheitsanalyse statt. Hierzu wurden aufbauend auf den Anforderungen aus Lasten- und Pflichtenheft zunächst Personas und Nutzer-Kontext-Szenarien abgeleitet. Die Persona-Beschreibungen gewährleisteten eine adressatengerechte Entwicklung der Benutzeroberfläche, welche die Anforderungen der potentiellen Anwender erfüllt. Die Nutzer-Kontext-Szenarien dienten der Analyse der benötigten Funktionen und Möglichkeiten seitens der Benutzeroberfläche.

Aufbauend auf diesen Ergebnissen wurden erste Wireframes erstellt, welche den grundlegenden Aufbau der Benutzeroberfläche darstellten. Diese dienten als Diskussionsgrundlage und waren die Basis für die weitere konzeptionelle Ausarbeitung der Benutzeroberfläche.

In einem weiteren Schritt wurden neben den einzelnen Interaktionen mit dem Funktionsmuster auch design-relevante Themen definiert. Im Rahmen von Mockups und Storyboards wurden die Abläufe aus den Nutzer-Kontext-Szenarien innerhalb der Benutzeroberfläche veranschaulicht. Mockups dienten der detail-und maßstabsgetreuen Visualisierung der Oberflächen.

Alle bis dahin erstellten Ergebnisse flossen in die Erstellung eines Styleguides für die Benutzeroberfläche ein, welcher als Grundlage für die Implementierung einer Benutzeroberfläche angesehen werden kann. Im Styleguide sind alle Gestaltungsrichtlinien (Farben, Schriften, etc.) für das Funktionsmuster festgelegt.

Die anschließende Realisierung der definierten Konzepte und die Anbindung der implementierten Benutzeroberfläche an das Gesamtsystem verliefen effizient und reibungslos. Durch ständige Reviews der Projektpartner war eine stetige Qualitätssicherung der Benutzeroberfläche gewährleistet.

10 Gesamtintegration des Funktionsmusters (AP 14)

Die Gesamtintegration des Funktionsmusters bedeutete die Verbindung der Ergebnisse der implementierungsträchtigen Arbeitspakete zu einer Software. Diese wird in diesem Dokument als Funktionsmuster bezeichnet. Die Integration der Ergebnisse erfolgte, wie geplant, nach und nach im Projektverlauf. Dabei traten keine größeren Probleme auf, und es mussten auch keine Änderungen an der Spezifikation des Funktionsmusters durchgeführt werden. Insgesamt wurden die Ergebnisse aus den AP 7, 8, 10, 11 und 12 zusammengeführt.

11 Qualitätssicherung (AP 15)

Die Qualitätssicherung erfolgte teilweise bereits im Rahmen der einzelnen AP. Bereits während der Implementierungsphase wurde der Quellcode dokumentiert. Zum Abschluss des Projekts wurde nochmals die gesamte Dokumentation überarbeitet und ergänzt. Des Weiteren wurden die Konzepte des Funktionsmusters dokumentiert, um eine einfache Verwertung und Erweiterung des Funktionsmusters zu ermöglichen. Zusätzlich wurde eine Bedienungsanleitung in Form einer Hilfe innerhalb des Funktionsmusters realisiert. Diese unterstützt eine bedienende Person bei ihren ersten Schritten im Umgang mit dem Funktionsmuster und bei Problemen. Eine ständige Aktualisierung bzw. Anpassung der Wissensbasis stellte zudem die Korrektheit der Wissensbasis sicher.

12 Test und Verifikation des Prototyps (AP 16)

Die Überprüfung der Funktionalität des Funktionsmusters erfolgte bereits während der Implementierung der einzelnen Funktion und wurde zum Projektende nochmals durchgeführt. Um die Schnittstellen des Web-Service und die dahinterliegenden Funktionalitäten zu testen, wurde ein Testwerkzeug entwickelt. Bei dem Testwerkzeug handelt es sich um eine lokale Client-Anwendung, die für jede Schnittstelle einen Aufruf-Button bereitstellt und ggf. benötigte Daten von dem Benutzer abfragt und das Ergebnis des Schnittstellenaufrufs ausgibt. Des Weiteren erfolgte der Test der Wissensbasis auf Korrektheit, zum einen durch die implementierte Validierungsfunktion und zum anderen durch eine manuelle Überprüfung sowie durch die Überprüfung der korrekten Arbeitsweise der Problemlösungskomponente.

Hierzu ist eine Beispielanlage, die auch Gegenstand und Vorlage für die IT-Sicherheitsanalyse war, detailliert manuell auf Grundlage des in diesem Projekt erstellten Bedrohungskataloges analysiert worden. Dabei ist ein Maximalprinzip dergestalt angewendet worden, dass eine mögliche Bedrohung auf jeden Fall zutrifft. Beispiel:

- Eine Software ist ungepatcht – also eine Schwachstelle vorhanden.
- Ein unsicheres Protokoll wird eingesetzt.
- Eine Regelung wird nicht eingehalten.

Alle auf diese Weise ermittelte Bedrohungen ziehen automatisch entsprechende Maßnahmen nach sich. Die so ermittelten Bedrohungs- und Maßnahmenlisten sind mit den Ergebnissen der Toolanalyse verglichen worden. Hierbei konnte durch mehrfache Analyse eine zufriedenstellende Annäherung zwischen manuellem Erstellen und automatisch erzeugten Ergebnissen erzielt werden.

Zudem wurde weiterhin die automatische Bestimmung der Security Level des Tools auf Korrektheit überprüft. Dazu wurden sämtliche Kombinationen, der fünf Fragen mit den jeweils drei

Antwortmöglichkeiten, mit den Ergebnissen des Tools in eine Wertetabelle eingetragen. Die Ergebnisse wurden mit den zu erwarteten Ergebnissen laut Regelwerk verglichen.

Die Benutzungsoberfläche wurde durch verschiedene Projektpartner auf ihre Funktionalität und Bedienbarkeit überprüft.

13 Vergleich der angestrebten und erreichten technischen Parameter

Zu Beginn des Projekts wurden im Rahmen des Lastenhefts Anforderungen an das zu entwickelnde Funktionsmuster formuliert. Im Folgenden soll ein Überblick über die (nicht-)funktionalen Anforderungen gegeben werden. Außerdem werden alle nicht (vollständig) umgesetzten Anforderungen aufgeführt und begründet, weshalb eine Anforderung nicht oder nur teilweise umgesetzt wurde. Das Lastenheft, welches als Grundlage dient, kann auf Wunsch nachgereicht werden.

13.1 Funktionale Anforderungen

Tabelle 8: Übersicht über alle (nicht) umgesetzten funktionalen Anforderungen

Anforderungsart	**Anzahl gesamt**	**Anzahl nicht (vollständig) erfüllt**
Funktionen des Funktionsmusters	19	3
Anforderungen an die Schnittstellen zum Informationsgewinn	3	1
Anforderungen an die Speicherung von Daten	7	1

13.2 Erläuterungen zu den nicht (vollständig) erfüllten funktionalen Anforderungen

Tabelle 9: Erläuterungen zu nicht umgesetzten Anforderungen bei Funktionsmusterfunktionen

Funktionen des Funktionsmusters		
Inhalt der Anforderung	**Umsetzungsgrad**	**Begründung**
Sollten für die IT-Sicherheitsanalyse Informationen notwendig sein, die über den Inhalt des automatischen Datenimports hinausgehen, muss das Funktionsmuster dem Benutzer erlauben die für die IT-Sicherheitsanalyse zusätzlich benötigten erweiterten Anlageninformationen über eine entsprechend gestaltete Oberfläche einzugeben.	Fast vollständig, Es können alle bestehenden Anlageninformationen bearbeitet sowie gelöscht werden. Lediglich komplett neue Assets können nicht manuell eingefügt werden.	Die Umsetzung des Hinzufügens neuer Assets wäre zu komplex. Zudem ist die Funktion für das Aufzeigen der Funktionalität des Funktionsmusters nicht notwendig.
Das Funktionsmuster soll die Möglichkeit bieten die abgeleiteten Schutzmaßnahmen für die spätere Verwendung innerhalb des Funktionsmusters zu speichern. Die Speicherung erfolgt dabei in einer eigenen Datei.	Anders umgesetzt	Im Rahmen des Pflichtenhefts wurde beschlossen, alle zu einem Projekt gehörigen Daten in einer Datei abzuspeichern. Deshalb werden Schutzmaßnahmen als Teil des Ergebnisses abgespeichert.
Das Funktionsmuster sollte eine Onlineaktualisierung von	Vorbereitet,	

IT-Security-Wissen und Industrial Ethernet Protokollen über das Internet ermöglichen. Auf diese Weise ist eine einfache Aktualisierung der Wissensbasis schnell möglich. Außerdem werden so auch aktuelle Bedrohungen und Schwachstellen erkannt.	In dem Funktionsmuster ist keine Server-Schnittstelle hinterlegt. Die Import-Funktion für eine Aktualisierung ist jedoch implementiert und funktionstüchtig.	

Tabelle 10: Erläuterungen nicht umgesetzte Anforderungen an Informationsgewinnschnittstellen

Anforderungen an die Schnittstellen zum Informationsgewinn		
Inhalt der Anforderung	**Umsetzungsgrad**	**Begründung**
Es wird eine Schnittstelle benötigt, die sich über das Internet mit dem Hersteller verbindet um das IT-Security-Wissen zu aktualisieren und neue Industrial Ethernet Protokolle hinzuzufügen.	Vorbereitet, In dem Funktionsmuster ist keine Server-Schnittstelle hinterlegt. Die Import-Funktion für eine Aktualisierung ist jedoch implementiert und funktionstüchtig.	

Tabelle 11: Erläuterungen zu den nicht umgesetzten Anforderungen an die Speicherung von Daten

Anforderungen an die Speicherung von Daten		
Inhalt der Anforderung	**Umsetzungsgrad**	**Begründung**
Bereits zu Beginn des Projektes wurde festgelegt, dass das Projektierungsmuster bestehende Anlageninformationen importieren können muss. Darunter auch die Projektierungsdaten aus dem Engineering-Werkzeug. Da diese Daten bereits an anderer Stelle vorliegen soll das Funktionsmuster keine zusätzliche Speicherung durchführen. Allerdings sollte das Funktionsmuster die Pfade der Datenquellen der letzten Sitzungen speichern, um dem Benutzer die Bedienung zu erleichtern.	Nicht umgesetzt	Wurde als nicht sinnvoll erachtet, da manuelle Bearbeitung der Anlageninformationen möglich ist.

13.3 Nicht funktionale Anforderungen

Tabelle 12: Übersicht über alle (nicht) umgesetzten nicht-funktionalen Anforderungen

Anforderungsart	Anzahl gesamt	Anzahl nicht (vollständig) erfüllt
Technische Restriktionen	2	0
Qualitative Anforderungen	6	0
Leistungsanforderungen	3	0
Anforderungen an Konfiguration und Wartung	1	0
Auswirkungen auf automatisierte Anlagen	1	0
Anforderungen an die Sicherheit	3	0
Entwicklungsrestriktionen	7	1

13.4 Erläuterungen zu den nicht (vollständig) erfüllten nicht-funktionalen Anforderungen

Tabelle 13: Erläuterungen zu den nicht umgesetzten Entwicklungsrestriktionen

Entwicklungsrestriktionen		
Inhalt der Anforderung	**Umsetzungsgrad**	**Begründung**
Die Kommunikationsverbindung zwischen der Anwendung und der Benutzeroberfläche soll mit Hilfe von Web-Services realisiert werden. Die Benutzeroberfläche selbst soll in HTML 5 und JavaScript entwickelt werden. Um die Benutzeroberfläche nutzen zu können werden folgende Browser mit mindestens der angegeben Version benötigt: - Internet Explorer 9.0+ - Firefox 3+ - Chrome 4.0+ - Safari 3.2+ - Opera 9.0+	Anforderung wurde abgeändert, Die Mindestversion der Browser wurde folgendermaßen angepasst: - IE ab 10 - Firefox ab 3.6 - Chrome ab 13 - Safari ab 6 - Opera ab 11.1	Wie sich herausstellte, unterstützen die Browser in der ursprünglich angegebenen Version die benötigte „File-API“ nicht, sodass die Versionsvorgaben für die Browser angepasst werden mussten.

14 Schlusswort

Das Funktionsmuster reduziert den Aufwand und das benötigte Wissen für IT-Sicherheitsanalysen gerade für KMU. Die Import-Funktion erlaubt die Aufnahme von Engineering-Daten als Eingangsinformation, sodass nur wenige manuelle Eingaben notwendig sind. Der Einsatz eines Regel-basierten Systems in Verbindung mit der Wissensbasis und der Update-Funktion reduzieren den Aufwand für KMU als Anwender weiter, da das Wissen automatisch angewendet wird. Die Ergebnisse einer IT-Sicherheitsanalyse können bei Bedarf überprüft werden, sodass der Anwender die vorgeschlagenen Schutzmaßnahmen nachvollziehen kann. Eine Umsetzung von Schutzmaßnahmen muss weiterhin manuell erfolgen.

Die Architektur des Funktionsmusters erlaubt einen Betrieb im eigenen Unternehmen, oder als Dienstleistung, wobei der Dienstleister die Pflege des IT-Sicherheitswissens übernimmt und bei Bedarf auch bei der Durchführung und Auswertung einer IT-Sicherheitsanalyse unterstützen kann.

Die durchgeführte Befragung als Teil der Anforderungsanalyse erlaubt die Ausrichtung des Projekts INSA auf die Bedürfnisse von KMU. Die Überprüfung des Funktionsmusters stellt die korrekte Durchführung von IT-Sicherheitsanalysen und Anwendung des Wissens sicher. Damit ist das Projekt INSA ein Schritt in Richtung jederzeit aktueller IT-Sicherheitskonzepte, bei gleichzeitiger Aufwandsreduzierung für KMU.

15 Veröffentlichungen

Im Rahmen des Projekts wurden einige Veröffentlichungen erstellt, die die Neuartigkeit des erarbeiteten Funktionsmusters und der dafür eingesetzten Konzepte, Technologien und Vorgehensweisen vorstellen. Zudem wurde mittels Presseberichten von verschiedenen Projektpartnern über das Projekt informiert. Zusätzlich konnte das Projekt INSA auf dem Infotag „Industrial Security“ des VDMA vorgestellt werden.

- Pressemitteilung zum Projektstart
- Vorstellung des Projekts INSA ihm Rahmen des Infotags „Industrial Security“ des VDMA in Frankfurt am Main am 10.20.2015.
- Tebbe, C., Glawe, M., Scholz, A., Niemann, K.-H., Fay, A.: Wissensbasierte Sicherheitsanalyse in der Automation: Aufwandsreduzierung am Beispiel der IT-Sicherheit. ATP Edition (4), S. 56–66, 2015.
- Glawe, M., Tebbe, C., Schewe, F., Fay, A., Niemann, K.-H.: Wissensbasierte Methoden zur Erstellung von IT-Sicherheitsanalysen automatisierter Anlagen. In: Verein Deutscher Ingenieure e.V. (Hrsg.): Automation 2015. Berlin: VDI-Verl., 2015.
- Tebbe C., Glawe, M., Dittgen, J., Niemann, K.-H., Fay, A.: Sicherheitslücken automatisch erkennen. In IEE Elektrische Automatisierung und Antriebstechnik. Hüthig Verlag, 60. Jahrgang, Heft 8-9/2015. S. 26-28. ISSN 1434-2898.
- Glawe, M., Tebbe, F., Fay, A., Niemann, K.-H.: Knowledge-based engineering of automation systems using ontologies and engineering data. In SciTePress (Hrsg.): Proceedings of KEOD 2015 – International Conference on Knowledge Engineering and Ontology Development, S. 1–8, 2015.
- Pressemitteilung zum Projektende

16 Literatur

[1] Allianz für Cyber-Sicherheit, *LARS ICS: Ein Werkzeug für den leichtgewichtigen Einstieg in industrielle Cyber-Security.* [Online]. Verfügbar unter: https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_downloads/materialien/tools/140627_LARS_ICS_Light_and_Right.html. Zugriff am: 22. Mai 2015.

[2] Department of Homeland Security, *CSET: Cyber Security Evaluation Tool.* [Online]. Verfügbar unter: https://ics-cert.us-cert.gov/Assessments. Zugriff am: 27. Juli 2015.

[3] ICS-CERT, *ICS-CERT: Industrial Control System - Cyber Emergency Response Team.* [Online]. Verfügbar unter: https://ics-cert.us-cert.gov/. Zugriff am: 31. März 2015.

[4] W3C, *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. W3C Recommendation. [Online]. Verfügbar unter: https://www.w3.org/TR/wsdl20/. Zugriff am: 24. April 2020.

[5] AutomationML e.V., *AutomationML.* [Online]. Verfügbar unter: https://www.automationml.org. Zugriff am: 22. Mai 2015.

[6] *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*, IEC 62424, 2008.

[7] *Security for Industrial Automation and Control Systems - Models and Concepts*, ISA-62443-1-1, 2015.

[8] A. Schüller, A. Scholz, T. Tauchnitz, R. Draht und T. Scherwietes, „Speed-Standardisierung am Beispiel der PLT-Stelle: Datenaustausch mit dem Namur-Datencontainer“, *atp edition - Automatisierungstechnische Praxis*, S. 36–46, Jan. 2015, 2015.

[9] W3C OWL Working Group, *OWL 2 Web Ontology Language: Document Overview.* [Online]. Verfügbar unter: http://www.w3.org/TR/owl2-overview/.

[10] I. Horrocks *et al., SWRL: A Semantic Web Rule Language Combining OWL and RuleML: W3C Member Submission.* [Online]. Verfügbar unter: http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/. Zugriff am: 10. Mai 2015.

[11] Deutsche Bibliothek, „Einführung in die XML Codierung von RDF: RDF in wissenschaftlichen Bibliotheken“, 2000. [Online]. Verfügbar unter: http://www.iwi-iuk.org/seminarNotes/1/RDF_xml.pdf. Zugriff am: 27. April 2015.

[12] bpellens, *owldotnetapi.* [Online]. Verfügbar unter: https://github.com/bpellens/owldotnetapi. Zugriff am: 10. Mai 2019.